Self-Supervised 3D Hand Pose Estimation Through Training by Fitting

Self-supervised 3D hand pose estimation through training by fitting

Chengde Wan[1], Thomas Probst[1], Luc Van Gool[1,3], and Angela Yao[2]

[1]ETH Zürich [2]National University of Singapore [3]KU Leuven

Abstract

We present a self-supervision method for 3D hand pose estimation from depth maps. We begin with a neural network initialized with synthesized data and fine-tune it on real but unlabelled depth maps by minimizing a set of data-fitting terms. By approximating the hand surface with a set of spheres, we design a differentiable hand renderer to align estimates by comparing the rendered and input depth maps. In addition, we place a set of priors including a data-driven term to further regulate the estimate's kinematic feasibility. Our method makes highly accurate estimates comparable to current supervised methods which require large amounts of labelled training samples, thereby advancing state-of-the-art in unsupervised learning for hand pose estimation.

1. Introduction

Deep learning has significantly advanced state-of-the-art for 3D hand pose estimation. The improvement in the accuracy of learning-based approaches can be attributed to two factors: choices in network design, and increased amounts of labelled data [29, 49, 53]. However, acquiring accurate 3D hand pose labels can be extremely difficult; current methods require marker-based motion capture [13], 6DoF sensors [53], or multi-view model-based tracking [45]. In all of these cases, careful supervision and manual cleaning of the labels is additionally necessary for high quality annotations. A commonly proposed alternative is to synthesize training samples – this is much easier, comes at virtually no cost and a variety of viewpoints, poses, and hand shapes can be generated. Unfortunately, the domain shift in terms of appearance, hand shape, and pose from synthesized to real data samples is highly non-trivial. As we show in our experiments, models trained on synthetic data exhibit a significant drop in accuracy when applied to real data.

A recent trend in hand pose estimation is to combine the benefits of learning-based discriminative approaches with model-based tracking methods [2, 13, 33, 43]. Model-based tracking casts pose estimation as a frame-wise model-fitting problem and requires no training data. Learning-based approaches can then be incorporated either by providing better initialization [33, 43], or by supporting the fitting with key-point estimations [2, 13] to reduce the risk of getting trapped in local minima.

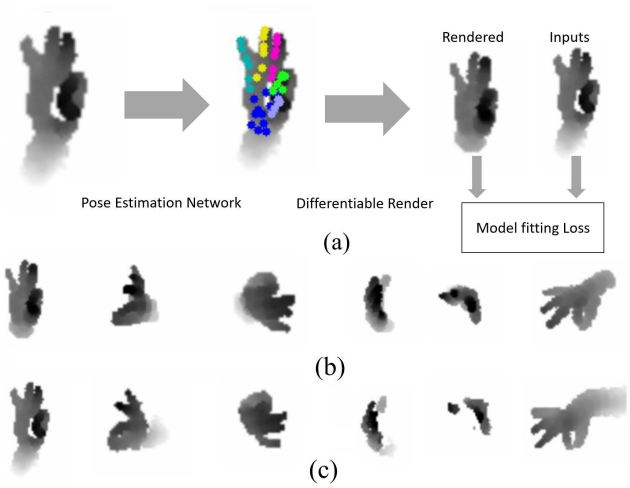

Figure 1. **Method overview.** (a) Outline of the proposed method; (b) Rendered depth map from estimation through differentiable renderer; (c) Corresponding input depth maps.

In this paper, we provide an alternative way to exploit the complementary benefits of unsupervised model-fitting and data-driven learning approaches. Specifically, we propose a self-supervision method for learning 3D hand pose from depth maps. Note that our approach does not require *any* manual annotation, either from labelling key-points [24, 54] or from manual initialization of a model-based tracker [40, 45]. At the same time, we are able to achieve accuracy comparable with current state-of-the-art that requires large amounts of labeled data.

Our proposed method supervises itself with a set of carefully designed differentiable model-fitting terms. Following a discriminative paradigm, hand poses are estimated with a deep neural network. During training, the network learns to make more accurate estimates by back-

propagating model-fitting errors that update network parameters. Unlike conventional model-based tracking, the hand poses in our method are not optimized in each frame independently; instead, the model-fitting error is minimized jointly over a large set of unlabelled images. As training progresses, the network learns to generalize from previous optimization results. We additionally feed the network with synthesized labelled data to avoid local minima and regularize the learning process. Interestingly, although we train without any (human-provided) labels, our method exhibits behaviour similar to traditional supervised methods: the accuracy steadily improves with increasing amounts of available data. For inference, our method is highly efficient, with only one forward pass through the network. In contrast, model-based tracking may need several optimization steps, especially in scenes with fast motions and large movements.

Our proposed model-fitting term penalizes the distance between the estimated hand surface to 3D points from the input depth map. For efficiency, we approximate the hand surface with a set of spheres (see Fig. 2) as used in [30]. This enables a fast and differentiable computation of the surface projection. For additional supervision, we train our network under a multi-view setup and apply a consistency loss term to overcome the ambiguities of self-occlusion. Finally, we penalize infeasible joint configurations by applying a variational auto-encoder (VAE) based prior.

We observe that previous methods which parameterize poses with joint angles [6, 55] tend to be sensitive to errors in parent node estimation. Further difficulty is introduced if one attempts to solve these angles via regression. As such, we directly estimate the 3D coordinates of the sphere centers. This conveniently allows us to work within a detection framework of fully convolutional networks (FCNs), which are used in many state-of-the-art methods [14, 41] and are more accurate than regression approaches. In contrast,

Our contributions can be summarized as follows:

- We propose a self-supervised method for 3D hand pose estimation from depth maps. Without any manual labels, the method achieves results comparable to state-of-the-art that requires large amounts of annotation.
- We propose a novel approach to couple unsupervised model-based fitting with supervised discriminative approaches for hand pose estimation.
- We provide a way to regularize kinematic feasibility in FCNs by placing a set of carefully designed priors, including a data-driven term learned by a VAE.

2. Related Works

Discriminative approaches. As a general trend, ever deeper and more sophisticated neural network architectures are dominating hand pose estimation methods. They are highly accurate [4, 5, 9, 11, 12, 20, 23, 31, 50] when trained with large amounts of labeled samples. However, given that accurate 3D annotations are extremely difficult to obtain, a number of works approach the problem with deep generative models to leverage unlabelled data [1, 3, 21, 28, 29, 36, 49]. Synthesizing depth maps ensures accurate annotations and seem to be a promising alternative but methods that rely only on synthesized data suffer from the large domain shift and actually perform much worse than when trained on less accurate real data [31, 34, 6]. To reduce the domain gap, Rad *et al.* [31] have proposed a domain adaptation method that tries to minimize feature differences from synthesized versus real images.

Simulated and unsupervised approaches. One line of work [6, 34] considers the more challenging setting of not using any manual labels. These models are trained on labeled but synthesized data and unlabeled real data. Shrivastava *et al*. [34] train a generative adversarial network with unlabelled depth maps to augment synthetic inputs with more realistic noise patterns. While the synthesized images better resemble images from real depth cameras, the domain gap beyond appearance, especially in hand pose and shape remain unsolved.

Dibra *et al*. [6] fine-tune a network trained on synthesized depth maps with unlabeled real data by minimizing a differentiable model-fitting error. While the fine-tuning improves the initial network accuracy, current state-of-the-art still outperforms the fine-tuned network by a large margin. Our approach is similar to [6] in that we also initialize the network with synthetic data and fine-tune with a data-fitting error. However, our method has several key differences including the hand model and model parameterization. We discuss the differences in detail in Section 4.3.

Hybrid approaches. Conventional model-based trackers incorporate discriminative models either for robust initialization [33, 43] or to augment the observation-based fit [2, 37]. Tompson *et al*. [45] estimates accurate hand poses with an expensive offline tracker and trains a network with those estimations for efficient online inference. More recently, focus has shifted to the integration of differentiable kinematic model representations such as forward kinematics [6, 16, 55] and linear blend skinning [6] into neural networks for end-to-end training.

Self supervision. A growing body of work tries to train neural networks without any human supervision. Common strategies include leveraging spatial or temporal context [7, 27], colour [18, 48], alignment [8] and privileged information from other modalities[26]. Recently, supervision from multiple views has been applied successfully to object key-point discovery [42], 3D reconstruction [47, 46], body pose estimation [32], and hand pose estimation [35]. We follow this line of work using multi-view supervision to resolve self-occlusions of the hand. In addition, beyond

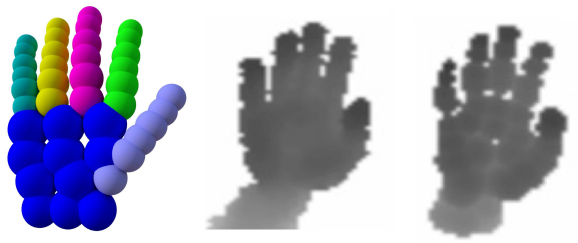

Figure 2. **Hand model used during model fitting.** Left: Our hand model approximation with 41 spheres; Middle: Real depth map; Right: Rendered depth map from differentiable ray-tracing.

enforcing multi-view consistency as in [32, 35], our differentiable depth renderer enables a dense and more detailed consistency error term by projecting hypotheses from one viewpoint to another.

3. Method

3.1. Hand model

We approximate the 3D hand surface with N=41 spheres (11 for the palm, 6 for each finger, see Figure 2(a)) similar to [30, 38, 39]. The sphere model $\mathcal{M}$ is parameterized as $\mathcal{M} = \{m^{(0)} \cdots m^{(N)}\}$, where $m^{(i)} = (x^{(i)}, y^{(i)}, z^{(i)}, r^{(i)})$ is the i^{th} sphere centered at $(x^{(i)}, y^{(i)}, z^{(i)})$ with radius $r^{(i)}$. The radii are predefined and remain unchanged during training. To this end, hand pose estimation is formulated as estimating a set of 3D key point coordinates.

3.2. Pose estimation network

Given a depth map, the pose estimation network tries to estimate the 3D coordinates of all N sphere centers. To adapt the FCN for 3D coordinate estimation, we follow the strategy of [14, 41, 42]. More specifically, the FCN regresses a heatmap h_{2D} of the 2D projections of the 3D points and a latent depth map h_{depth} encoding the point's depth information. The 3D coordinates are then recovered by integrating over the heatmap and latent depth map:

$$(x, y) = \sum_{x_i} \sum_{y_i} (x_i, y_i) \text{softmax}(\alpha h_{2D})(x_i, y_i) \quad (1)$$

$$z = \sum_{x_i} \sum_{y_i} h_{depth}(x_i, y_i) \text{softmax}(\alpha h_{2D})(x_i, y_i) \quad (2)$$

where x_i and y_i are the ith pixel's coordinates on the 2D grid and α is the annealing factor. We set as $\alpha = 10$ in rest of the paper. The softmax function serves as an approximation to argmax. We refer the reader to [14, 41, 42] for more details. We use the hourglass network[22] as our FCN architecture.

3.3. Network initialization

We initialize the network by training on a synthesized dataset. Depth maps are synthesized from the hand model provided by [45] according to the sampling strategy of [33] for generating random hand poses. More details are provided in the Supplementary Materials. During training, 20 million synthesized depth maps with different poses and view points are generated online and fed to the network.

The training loss is formulated as a mean squared error between the estimated 3D coordinates integrated from h_{2D} and h_{depth} (see Equations 1 and 2), and the ground truth coordinates, similar to [14, 19, 41]. Alternatively, as done in [42], supervision can be provided directly at the level of h_{2D} and h_{depth}. The j^{th} 3D ball's label for $h_{2D}^{(j)}$ is a 2D heatmap centered around the 2D projection of the ball. $h_{depth}^{(j)}$ at the v^{th} column and the u^{th} row is generated as

$$h_{depth}^{(j)}(v, u) = \begin{cases} d_j & \text{if} \quad h_{2D}^{(j)}(v, u) > 0, \\ 0 & \text{others;} \end{cases} \quad (3)$$

where d_j is the depth of the j^{th} joint. In preliminary experiments we found that both strategies perform similarly and for the rest of the paper, we use the latter strategy to be analogous to [42].

3.4. Self-supervision by model fitting

For a given depth map of a hand, we begin by estimating the spheres' center coordinates. From these coordinates we render the spheres and evaluate how well the they fit with the input depth map according to an energy function. We use a differentiable rendering process that allows back-propagation of errors from the energy function for gradual fine-tuning of the estimations.

At first glance, our overall process resembles conventional model-based tracking. However, our method is fundamentally different in that we are optimizing over neural network parameters rather than pose parameters. We share the benefits of data-driven approaches because we minimize the model-fitting error over an entire set of unlabeled depth maps, rather than fitting frames independently as done in model-based tracking. As shown in our experiments (see section 4.2), the joint optimization over a set of data gives rise to accuracy improvements when the set size increases. In this context, the model-fitting energy can be directly interpreted as a self-supervised training loss. The trained network generalizes from previously estimated samples while still leveraging supervision from synthesized labeled data. Moreover, the method enables efficient inference, using only one forward pass, whereas model-based tracking needs initialization and multiple iterations of optimization.

The self-supervised training loss, composed of data and

prior terms, is defined as follows,

$$L(\theta) = \underbrace{L_{\text{m2d}}(\theta) + \lambda_1 L_{\text{d2m}}(\theta) + \lambda_2 L_{\text{multiview}}(\theta)}_{\text{data terms}} + \underbrace{\lambda_3 L_{\text{vae}}(\theta) + \lambda_4 L_{\text{bone length}}(\theta) + \lambda_5 L_{\text{collision}}(\theta)}_{\text{prior terms}}, \quad (4)$$

where θ are network parameters.

3.4.1 Data terms

The **model to data term** L_{m2d} aligns the spheres as close as possible to the surface points in the depth map. We define it as the L_1 distance between the input depth map $\mathcal{D}_i$ and the rendered depth map

$$L_{\text{m2d}}(\theta) = \sum_i \left| G(f(\mathcal{D}_i|\theta)) - \mathcal{D}_i \right|, \quad (5)$$

where $f(\cdot|\theta)$ estimates the coordinates of sphere centers from input depth map and $G(\cdot)$ renders the estimate to a depth map. The $L1$ loss makes the training robust to holes in the input depth map. The rendered depth map is a composite of all the spheres rendered independently followed by a min pooling in the z dimension across the spheres. The min pooling serves as a depth buffer check in a standard rendering pipeline. For each pixel on the rendered image, the rendering process checks if the corresponding ray intersects with the sphere and calculates the depth at the point of intersection. We apply an orthographic projection for the rendering. The rays are perpendicular to the image plane, and the depth at pixel (u, v) for j^{th} sphere $m^{(j)}$ with radius $r^{(j)}$ centered at $(x^{(j)}, y^{(j)}, z^{(j)})$ *w.r.t.* the camera coordinate frame is calculated as

$$g(m^{(j)})_{uv} = \begin{cases} z^{(j)} - \sqrt{(r^{(j)})^2 - d^2(u,v)} & \text{if } d(u,v) \leq r, \\ d_{\text{far}} & \text{others}, \end{cases} \quad (6)$$

where

$$d(u, v) = \|(x^{(j)}, y^{(j)}) - \Pi^{-1}(u, v)\|_2 \quad (7)$$

measures the distance between the ray and sphere center and $\Pi^{-1}(u, v)$ is the inverse orthographic projection. The rendering process G of an estimated set of spheres $\mathcal{M}_i$ can be formulated as

$$G(\mathcal{M}_i) = \min_j g(m_i^{(j)}). \quad (8)$$

Note that the entire rendering process is fully differentiable and can be implemented easily with any deep learning framework. The rendered primitive depth map and its input depth map counterpart are shown in Fig.2.

The **data to model term** L_{d2m} is a registration loss between the estimated model and input depth map. Since there are no gradients on the background of the rendered depth map according to Equation 6, the modle to data term L_{m2d} alone cannot push the model towards unexplained points on the input depth map. This is taken care of by L_{d2m} which works in the spirit of ICP by minimizing the distance between every point p from the depth map $\mathcal{D}_i$ and its projection on to the estimated hand model surface $\mathcal{M}_i$,

$$L_{\text{d2m}}(\theta) = \sum_i \sum_{p \in \mathcal{D}_i} d(p, \Pi_{\mathcal{M}_i}(p)). \quad (9)$$

The distance $d(p, \Pi_{\mathcal{M}_i}(p))$ is estimated as the distance to every single sphere $m^{(j)}$ with radius $r^{(j)}$ centered at $c^{(j)}$ as

$$d(p, m^{(j)}) = \text{abs}(\|p - c^{(j)}\|_2 - r^{(j)}) \quad (10)$$

and takes the minimum among all balls as

$$d(p, \Pi_{\mathcal{M}_i}(p)) = \min_j d(p, m^{(j)}). \quad (11)$$

The **multi-view consistency term** $L_{\text{multiview}}$ provides supervision from multiple viewpoints; this mitigates the ambiguities and errors that arise from the frequent self-occlusion of hands. For training we assume a calibrated multi-camera set-up and maintain consistency between the viewpoints in two ways. Firstly, as shown in Figure 3, we project the centers of spheres estimated from view v_i to view v_j and evaluate L_{d2m} and L_{m2d} with respect to the depth map captured from view v_j. To further propagate estimations from different viewpoints, an additional multi-view term is defined as:

$$L_{\text{multiview}}(\theta) = \sum_i \sum_j \sum_v \|T_v(c_i^{(j,v)}) - \bar{c}_i^{(j)}\|_2^2, \quad (12)$$

where $c_i^{(j,v)}$ is the center of j^{th} sphere estimated from the i^{th} depth map with the v^{th} camera. $T_v(\cdot)$ projects points from view v to a canonical frame while $\bar{c}_i^{(j)}$ is the corresponding robust average from the multi-view estimations in the canonical frame. The robust average $\bar{c}_i^{(j)}$ can be either the median or a selected c_i^j based on the heat map with minimal variance. In preliminary tests, we found both strategies perform equally well and we use the median for our experiments. In contrast, simply averaging over the multiple views degrades results as the mean is sensitive to outliers.

3.4.2 Prior terms

Because we do not have a skeleton model, we cannot enforce conventional kinematic constraints such as joint angle ranges. As an alternative, we adopt a data-driven approach to encourage the estimated sphere positions to form a kinematically feasible pose. More specifically, we apply a **vae term** L_{vae} which aims to maximize the likelihood lower

bound of the hand pose configuration. Since the sphere positions inherently populate only a subspace, we train a variational auto-encoder (VAE)[17] over the estimated sphere centers to learn this latent space. The pose training samples of the VAE are generated by the same kinematics model and sampling strategy as used in Section 3.3 and is therefore unsupervised. We use the standard variational lower bound to ensure that the estimated pose parameters lie in the learned subspace as follows,

$$L_{\text{vae}}(\theta) = \sum_i E_{z\sim Q}[\log P\big(f(\mathcal{D}_i|\theta)|z\big)] - D_{\text{KL}}[Q(z)||P(z)], \tag{13}$$

where $f(\cdot|\theta)$ is the pose estimation network. We refer the reader to [17] for more details on learning and inference in the VAE. The VAE is trained in advance and its weights remain unchanged during training.

The **bone length term** $L_{\text{bone length}}$ ensures that distances between two bone end points remain unchanged:

$$L_{\text{bone length}} = \sum_{i,j,k} \max(d_i^{jk} - l_{\max}^{jk}, 0)^2 + \max(l_{\min}^{jk} - d_i^{jk}, 0)^2, \tag{14}$$

where

$$d_i^{jk} = \|c_i^{(j)} - c_i^{(k)}\|_2 \tag{15}$$

measures the estimated bone length with the estimated end points $c_i^{(j)}$ and $c_i^{(k)}$ from the i^{th} input. $\big[l_{\min}^{jk}, l_{\max}^{jk}\big]$ are predefined ranges based on different skeleton sizes.

The **collision term** $L_{\text{collision}}$ penalizes self collisions between the j^{th} and k^{th} sphere as follows,

$$L_{\text{collision}} = \sum_{i,j,k} \max(r^{(j)} + r^{(k)} - \|c_i^{(j)} - c_i^{(k)}\|_2^2, 0). \tag{16}$$

4. Experimentation

We evaluate our method on the NYU Hand Pose Dataset [45], which is currently the only publicly available multi-view depth dataset. The dataset, captured by 3 calibrated and synchronized PrimeSense depth cameras, consists of 72757×3 training frames and 8252×3 for testing. NYU is a challenging dataset with a wide coverage of hand poses. We apply the ground-truth annotation only to calculate camera extrinsics. In addition, we synthesize a dataset of 20K depth maps to evaluate how well the trained network can generalize to (new) synthesized samples.

We quantitatively evaluate with two standard metrics: mean joint position error (in mm) averaged over all joints and frames, and the percentage of successful frames, *i.e.* those in which all joint predictions are within a certain threshold [44]. Qualitative results of the estimated hand poses and their sphere model renderings from other viewpoints by the differentiable renderer are shown in Figure 3. By default, we report the result using a single stack hourglass network.

One should note that in our self-supervised scenario, the pose estimation error on the training set is fundamentally different than the training error in standard supervised training. In self-supervised learning, we are minimizing a data-fitting error, which *should* be correlated with pose estimation accuracy, but there are no guarantees that lower errors will give rise to more accurate pose estimates.

4.1. Training with only synthesized data

We first evaluate how a network trained on purely synthesized data generalizes to new synthesized samples and as well as real depth maps. Table 1 (synth, synth(test on test)) and Figure 5 (cyan dotted line) shows that this network is highly accurate on unseen synthesized samples; 78% of frames have maximum errors less than 20mm and the mean joint error is only 6.76mm. However, the accuracy deteriorates dramatically when testing on real-world depth maps – the mean average joint error increases almost four-fold to 27.85mm. If we augment the synthesized depth maps with random noise (synth, aug. with noise), we can reduce the mean joint error to 22.65mm. This shows that the neural network easily over-fits to certain local patterns; we speculate that it is likely rasterization artifacts on the synthesized depth maps. We use the network trained from synthesized depth maps augmented with noise as the initial network for subsequent experiments in this section.

4.2. Ablation studies

Impact of multi-view supervision. To what extent do multiple viewpoints help with self-supervision? We first design a single-view baseline (100% single view in Figure 6 and Table 1) without the multi-view consistency term and without projecting the pose estimates to other viewpoints. Since we use the median as the robust average in measuring multi-view consistency (see Equ. 12), using only 2 views is not applicable. In the subsequent experiments, we denote the setting of using 3 views, all prior terms, and the data fitting losses L_{m2d} and L_{d2m}, as 'multi-view'.

When comparing to the results of training under a multi-view set up (100% multi-view in Figure 6 and Table 1), accuracy degenerates for both mean joint error and percentage of successful frames within the error threshold below 50mm. This baseline shows that our current self-supervision strategy under a monocular setup is insufficient to resolve the hand's self-occlusions – which can be quite extreme. Yet when comparing with the initial results of training from synthetic data, there is significant improvement. The mean average error decreases from 22.65mm to 17.79mm (see Table 1), which validates the effectiveness of our single view's self-supervision terms.

Requiring multi-view setups can limit data capture scenarios; they cannot be applied to egocentric views and are hard to apply when the user moves within a large area. This

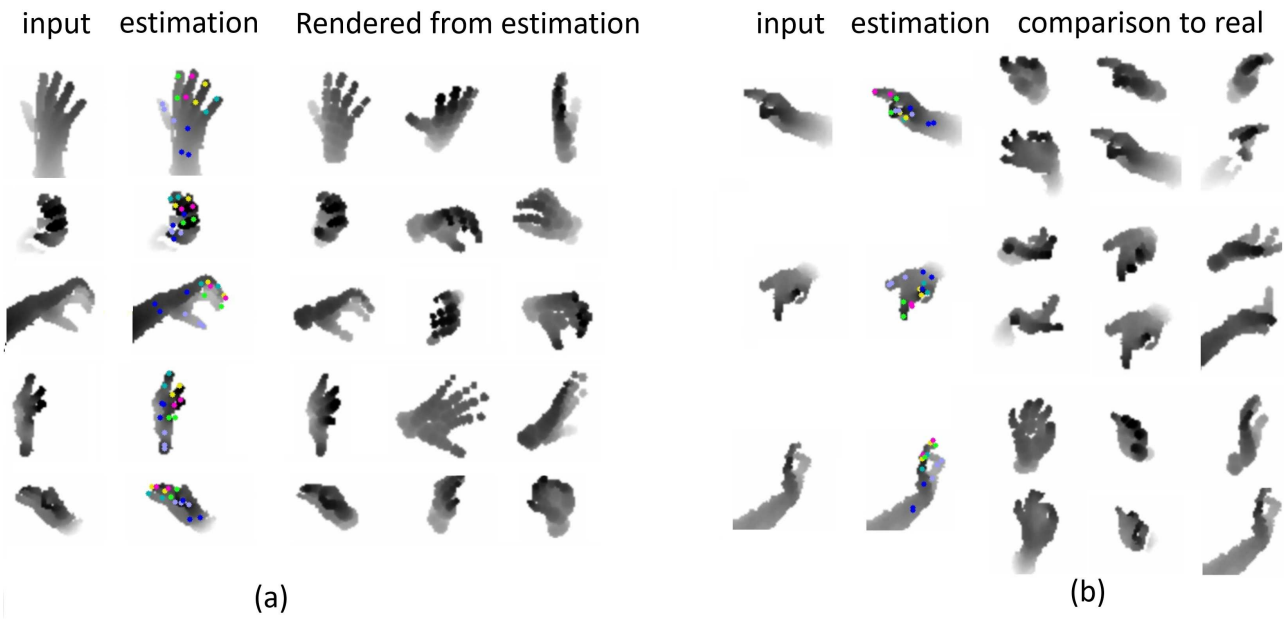

Figure 3. **Qualitative results.** (a) Success cases. 3 right columns: estimated poses rendered from different viewpoints by the differentiable renderer. (b) Failure cases. 3 right columns: rendered estimaties (top rows) from different views and corresponding depth map)

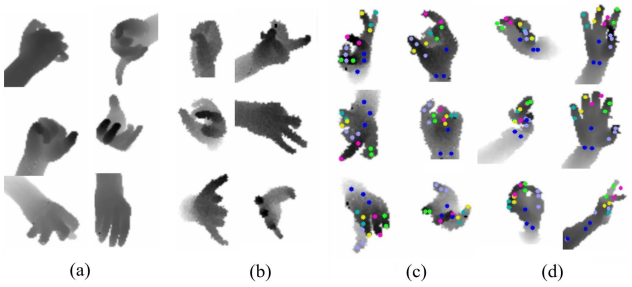

Figure 4. **Synthesized depth maps.**(a) Synthesized depth map from polygonal mesh (Sec. 3.3); (b) Synthesized depth maps augmented with random noise; (c) Estimation of unseen synthesized depth maps; (d) Estimation of real depth maps. The network is trained with synthesized data augmented with noise in (c) and (d).

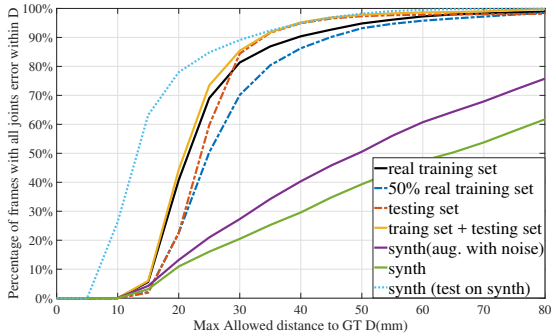

Figure 5. **Impact of training data.** Percentage of successful frames with different error thresholds, trained on different training set. The "synth (test on synth)" is tested on a 20k synthetic dataset and the rest methods are all tested on the NYU[45] test set.

begs the question whether we can design a more flexible learning scheme to use a mixture of both multi-view and single-view depth maps. To that end, we halve the dataset, where the first half has multiple views and the second half has only a single view and apply these to two additional baselines. The first baseline is trained only on the first half (50% multi-view) and while the second baseline is trained on both (50% multi-view + 50% single view). As shown in Figure 6, the additional single-view training data can improve the percentage of successful frames with threshold of 20mm from 22.5% to 35.0% and decreases the mean average error from 13.77mm to 13.33mm.

We conclude that multi-view set up is critical in self-supervision to resolve (self) occlusions. Meanwhile, our single-view self-supervision terms help improve accuracy, offering flexibility for setups where it is not possible to capture data from multiple views.

Impact of prior terms. We study the individual contributions of the three priors by training and testing the network without the L_{vae} ("without vae loss"), $L_{\text{collision}}$ ("without collision loss") and $L_{\text{bone length}}$ ("without bone length loss") terms. Figure 7 shows that without regulating the estimated pose with any of these priors, there is a dramatic decrease on the percentage of successful frames, especially in the range of error thresholds form 20mm to 40mm. This validates each of the priors in enforcing kinematically feasible pose estimates. The corresponding mean joint position error increases similarly, as shown in Table 1. Interestingly, the pose estimation is worse in the absence of $L_{\text{bone length}}$,

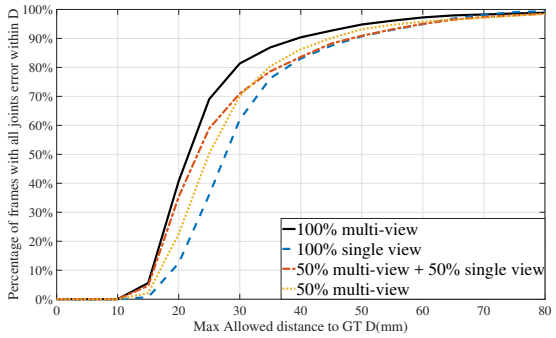

Figure 6. **Impact of multi-view supervision.** Percentage of successful frames within different error thresholds, trained on different training set. All methods are tested on the NYU [45] test set.

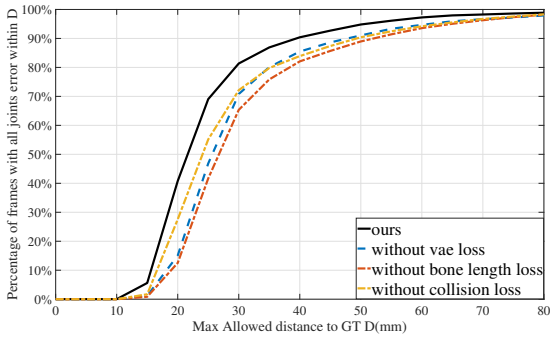

Figure 7. **Impact of prior terms.** The percentage of successful frames with different error thresholds, trained with different prior terms. All methods are tested on the NYU [45] test set.

even if it is still constrained by L_{vae}. In theory, the learned VAE should already encode the bone length constraints as part of the prior. This reveals that the L_{vae} term alone is insufficient to ensure kinematic feasibility. We note however, that the current prior terms are just that – they are priors. They cannot guarantee that pose estimates will meet strict kinematic constraints, *e.g*. the same subject has a constant bone length, or that joint angles do not exceed a predefined range. If this is desired, inverse kinematics can be added as a post processing step [45, 52].

Variation in training data. We investigate how different training data influences the resulting network with two baselines. First, we train only with the 8252×3 testing samples to check how well self-supervision might (over-)fit the network to training data. We then trained with a combination of both the testing and training samples. Finally, we compare these two setups with our conventional baseline of training and testing on the originally designated training and test splits.

Interestingly enough, training directly on the test samples alone results in a higher mean joint error than when training on the training samples (14.53mm vs 12.62mm error on the test samples, see Table 1). Similarly, the percentage of successful frames is only 22.5% as opposed to 40.7% at the 20mm error threshold (see Figure 5). We attribute this to the current model fitting terms and optimization with back-propagation; it cannot give rise to highly accurate pose estimates with only small amounts of training data. However, if the amount of training data increases, then so does the accuracy. These benefits justify a data-driven based self-supervision approach over conventional model-based tracking which optimizes each frame independently. Sure enough, when training with the combined training and test set, we further improve (marginally, since we only add about 10% more data), by decreasing mean joint position error from 12.62mm to 12.31mm and by increasing the percentage of successful frames from 40.7% to 44.3%.

Method	Mean joint error
ours (100% multiple view)	12.62 mm
100% single view	17.79 mm
50% multi-view	13.77 mm
50% multi-view + 50% single view	13.33 mm
without vae loss	14.17 mm
without bone length loss	14.56 mm
without collision loss	13.73 mm
50% multi-view	13.77 mm
50% multi-view + 50% single view	13.33 mm
synth(aug. with noise)	22.65 mm
synth	27.85 mm
synt(test on synth)	6.76 mm
train on test	14.53 mm
train on test + train	12.31 mm

Table 1. **Ablation study and self comparison.** We report mean joint error averaged over all joints and frames.

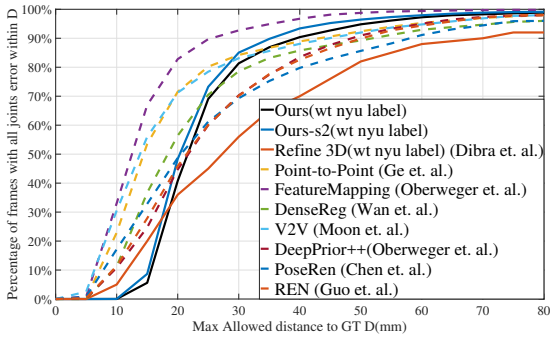

Figure 8. **Comparison to state-of-arts.** We report the result on NYU[45] testing set and plot the percentage of frames in which all joints are below a threshold.

4.3. Comparison to state-of-the-art

We first compare our result with, to the best of our knowledge, the only other unsupervised method [6]. Similar to us, [6] uses a CNN and pretrains their network with

synthesized depth maps. This followed by a model-fitting based fine-tuning. As can be seen in Figure 8, our network outperforms [6] by a large margin for the percentage of successful frames at error thresholds higher than 25mm. We achieve a much higher accuracy for two reasons. First, our method exploits the benefits of an FCN, while [6] directly regressed joint angles; secondly there are no gradients in their depth term (Eq. 6 in [6]) associated with unexplained points from the depth map which we handle with L_{d2m}.

Next, we compare our results to state-of-the-art [4, 10, 11, 12, 20, 23, 25, 49, 50, 51, 55] which train with ground-truth annotations. Surprisingly, our result outperforms all of them in terms of percentage of successful frames when thresholds are larger than 30mm. This again validates self-supervision and supports the possibility of learning robust and accurate pose estimation systems without labels.

One drawback of our current model is that under more stringent error criteria, *e.g*. when thresholds are 20mm or less, our accuracy is no longer as good as state-of-the-art. We attribute this to two causes. First, approximating the hand surface with only spheres is insufficient and cannot capture smaller fitting errors. To improve the accuracy, one will need to use finer models such as a more personalized hand mesh model though this comes at greater computational expense [2, 15]. Secondly, the current prior terms, because they do not place strict kinematic constraints, likely give rise to small offsets over the joints.

When comparing in terms of the average joint position error, our method falls short of current state-of-the-art. However, we note that the mean joint error, as a mean, can be slightly biased, in the sense that certain joints are "easier" to estimate. The finger roots, palm center and wrist, are less sensitive to larger offsets than the finger tips, even though the tips are more critical for good user experience in real-world applications. Actually, our finger-tip accuracy, with 11.77mm(ours) / 12.39mm(ours stack=2), is higher than the mean joint error once the roots, palm center and wrist joints are included (12.26mm(ours) / 12.62mm(ours stack=2)).

Currently, [31] reports the highest accuracy to date (see Table 2 by using domain adaptation techniques to leverage synthesized data together with labelled real data. Such a technique is complementary to our proposed self-supervision strategy; the two combined together are likely to lead to even higher accuracies in pose estimates.

Method	mean joint error
ours (stack = 1)	12.6 mm
ours (stack = 2)	12.3 mm
FeatureMapping [31]	7.4 mm
V2V [20]	8.4 mm
Point-to-Point[11]	9.0 mm
DenseReg [50]	10.2 mm
DeepPrior++ [23]	12.2 mm
Pose-REN [4]	11.8 mm
Ren [12]	12.7 mm
3DCNN [10]	14.1 mm
Lie-X [51]	14.5 mm
CrossingNet [49]	15.5 mm
Feedback [25]	15.9 mm
DeepModel [55]	17.0 mm

Table 2. **Comparison with state-of-the-art.** Mean joint error averaged over all joints and frames. All methods are tested on the NYU[45] test set.

5. Conclusion

We present a self-supervision method for 3D hand pose estimation from single depth maps. Our method does not require any manual annotation yet can produce highly accurate estimates with results competitive to current supervised state-of-arts. We formulate the training loss of the pose estimation network in terms of a data-fitting error whereby the hand surface is approximated with a set of spheres. By parameterizing the pose directly as the sphere centers, our method exploits the benefits of FCNs and avoids the difficulties of direct angular regression. In addition to data terms, we place priors, including a data-driven term from a trained VAE to encourage kinematic feasibility.

Through our model, we are able to jointly benefit from model-based tracking, which requires no supervision, and from data-driven approaches, in which the accuracy steadily improves given more training data, even without labels. In the future, we look forward to incorporating our unsupervised approach with domain adaptation methods to further improve the accuracy with available labelled data.

Acknowledgements The authors gratefully acknowledge support from ETH Computer Vision Lab's institutional funding, the Chinese Scholarship Council, and Singapore Ministry of Education's Academic Research Fund Tier 1.

References

[1] S. Baek, K. I. Kim, and T.-K. Kim. Augmented skeleton space transfer for depth-based hand pose estimation. In *CVPR*, 2018.

[2] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys. Motion capture of hands in action using discriminative salient points. In *ECCV*, 2012.

[3] Y. Cai, L. Ge, J. Cai, and J. Yuan. Weakly-supervised 3D hand pose estimation from monocular RGB images. In *ECCV*, 2018.

[4] X. Chen, G. Wang, H. Guo, and C. Zhang. Pose guided structured region ensemble network for cascaded hand pose estimation. *arXiv preprint arXiv:1708.03416*, 2017.

[5] X. Chen, G. Wang, C. Zhang, T.-K. Kim, and X. Ji. SHPR-Net: Deep semantic hand pose regression from point clouds. *IEEE Access*, 6:43425–43439, 2018.

[6] E. Dibra, T. Wolf, C. Oztireli, and M. Gross. How to refine 3d hand pose estimation from unlabelled depth data? In *3D Vision (3DV)*, 2017.

[7] C. Doersch, A. Gupta, and A. A. Efros. Unsupervised visual representation learning by context prediction. In *ICCV*, 2015.

[8] X. Dong, S.-I. Yu, X. Weng, S.-E. Wei, Y. Yang, and Y. Sheikh. Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors. In *CVPR*, 2018.

[9] L. Ge, Y. Cai, J. Weng, and J. Yuan. Hand PointNet: 3D hand pose estimation using point sets. In *CVPR*, 2018.

[10] L. Ge, H. Liang, J. Yuan, and D. Thalmann. 3D convolutional neural networks for efficient and robust hand pose estimation from single depth images. In *CVPR*, 2017.

[11] L. Ge, Z. Ren, and J. Yuan. Point-to-point regression pointnet for 3d hand pose estimation. *ECCV*, 2018.

[12] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang. Region ensemble network: Improving convolutional network for hand pose estimation. In *ICIP*, 2017.

[13] S. Han, B. Liu, R. Wang, Y. Ye, C. D. Twigg, and K. Kin. Online optical marker-based hand tracking with deep labels. *ACM Transactions on Graphics (TOG)*, 2018.

[14] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz. Hand pose estimation via latent 2.5D heatmap regression. In *ECCV*, 2018.

[15] D. Joseph Tan, T. Cashman, J. Taylor, A. Fitzgibbon, D. Tarlow, S. Khamis, S. Izadi, and J. Shotton. Fits like a glove: Rapid and reliable hand shape personalization. In *CVPR*, 2016.

[16] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik. End-to-end recovery of human shape and pose. In *CVPR*, 2018.

[17] D. P. Kingma and M. Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

[18] G. Larsson, M. Maire, and G. Shakhnarovich. Colorization as a proxy task for visual understanding. In *CVPR*, 2017.

[19] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. VNect: real-time 3D human pose estimation with a single rgb camera. 2017.

[20] G. Moon, J. Y. Chang, and K. M. Lee. V2V-PoseNet: voxel-to-voxel prediction network for accurate 3D hand and human pose estimation from a single depth map. In *CVPR*, 2018.

[21] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt. GANerated hands for real-time 3D hand tracking from monocular RGB. In *CVPR*, 2018.

[22] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *ECCV*, 2016.

[23] M. Oberweger and V. Lepetit. Deepprior++: Improving fast and accurate 3d hand pose estimation. In *ICCV workshop*, 2017.

[24] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit. Efficiently creating 3D training data for fine hand pose estimation. In *CVPR*, 2016.

[25] M. Oberweger, P. Wohlhart, and V. Lepetit. Training a feedback loop for hand pose estimation. In *ICCV*, 2015.

[26] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.

[27] D. Pathak, R. B. Girshick, P. Dollár, T. Darrell, and B. Hariharan. Learning features by watching objects move. In *CVPR*, 2017.

[28] G. Poier, M. Opitz, D. Schinagl, and H. Bischof. MURAUER: Mapping unlabeled real data for label austerity. In *WACV*, 2019.

[29] G. Poier, D. Schinagl, and H. Bischof. Learning pose specific representations by predicting different views. In *CVPR*, 2018.

[30] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *CVPR*, 2014.

[31] M. Rad, M. Oberweger, and V. Lepetit. Feature mapping for learning fast and accurate 3D pose inference from synthetic images. In *CVPR*, 2018.

[32] H. Rhodin, J. Spörri, I. Katircioglu, V. Constantin, F. Meyer, E. Müller, M. Salzmann, and P. Fua. Learning monocular 3D human pose estimation from multi-view images. In *CVPR*, 2018.

[33] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnikov, Y. Wei, et al. Accurate, robust, and flexible real-time hand tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, 2015.

[34] A. Shrivastava, T. Pfister, O. Tuzel, J. Susskind, W. Wang, and R. Webb. Learning from simulated and unsupervised images through adversarial training. In *CVPR*, 2017.

[35] T. Simon, H. Joo, I. A. Matthews, and Y. Sheikh. Hand keypoint detection in single images using multiview bootstrapping. In *CVPR*, 2017.

[36] A. Spurr, J. Song, S. Park, and O. Hilliges. Cross-modal deep variational hand pose estimation. In *CVPR*, 2018.

[37] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *CVPR*, 2015.

[38] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *ICCV*, 2013.

[39] S. Sridhar, H. Rhodin, H.-P. Seidel, A. Oulasvirta, and C. Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. In *3D Vision (3DV)*, 2014.

[40] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun. Cascaded hand pose regression. In *CVPR*, 2015.

[41] X. Sun, B. Xiao, S. Liang, and Y. Wei. Integral human pose regression. In *ECCV*, 2018.

[42] S. Suwajanakorn, N. Snavely, J. Tompson, and M. Norouzi. Discovery of latent 3D keypoints via end-to-end geometric reasoning. In *NIPS*, 2018.

[43] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton. Opening the black box: Hierarchical sampling optimization for estimating human hand pose. In *ICCV*, 2015.

[44] J. Taylor, J. Shotton, T. Sharp, and A. Fitzgibbon. The vitruvian manifold: Inferring dense correspondences for one-shot human pose estimation. In *CVPR*, 2012.

[45] J. Tompson, M. Stein, Y. Lecun, and K. Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (ToG)*.

[46] S. Tulsiani, A. A. Efros, and J. Malik. Multi-view consistency as supervisory signal for learning shape and pose prediction. *CVPR*, 2018.

[47] S. Tulsiani, T. Zhou, A. A. Efros, and J. Malik. Multi-view supervision for single-view reconstruction via differentiable ray consistency. In *CVPR*, 2017.

[48] C. Vondrick, A. Shrivastava, A. Fathi, S. Guadarrama, and K. Murphy. Tracking emerges by colorizing videos. In *ECCV*, 2018.

[49] C. Wan, T. Probst, L. Van Gool, and A. Yao. Crossing nets: Combining gans and vaes with a shared latent space for hand pose estimation. In *CVPR*, 2017.

[50] C. Wan, T. Probst, L. Van Gool, and A. Yao. Dense 3D regression for hand pose estimation. In *CVPR*, 2018.

[51] C. Xu, L. N. Govindarajan, Y. Zhang, and L. Cheng. Lie-X: Depth image based articulated object pose estimation, tracking, and action recognition on lie groups. *International Journal of Computer Vision (IJCV)*, 123(3):454–478, 2017.

[52] Q. Ye, S. Yuan, and T.-K. Kim. Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation. In *ECCV*, 2016.

[53] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T.-K. Kim. BigHand2.2M benchmark: Hand pose dataset and state of the art analysis. In *CVPR*, 2017.

[54] J. Zhang, J. Jiao, M. Chen, L. Qu, X. Xu, and Q. Yang. A hand pose tracking benchmark from stereo matching. In *ICIP*, 2017.

[55] X. Zhou, Q. Wan, W. Zhang, X. Xue, and Y. Wei. Model-based deep hand pose estimation. In *IJCAI*, 2016.